山东大学 计算机科学与技术 学院

操作系统 课程实验报告

|  |  |  |  |
| --- | --- | --- | --- |
| 学号：201705130113 | 姓名：黄瑞哲 | | 班级：计科17.3 |
| 实验题目：进程调度算法实验 | | | |
| 实验学时：2 | | 实验日期：2019.11.4 | |
| 实验目的：  加深对进程调度概念的理解，体验进程调度机制的功能，了解 Linux 系统中进程 调度策略的使用方法。 练习进程调度算法的编程和调试技术。 | | | |
| 硬件环境：  Intel Core i5-8300H | | | |
| 软件环境：  Windows 10 Pro 1903  Docker for Windows  Ubuntu 18.04  VS Code | | | |
| 实验内容与设计：    **获取进程动态优先数的系统调用语法为**      对于示例实验，程序完成调度策略和优先级的设置，这里要注意的一点是调整调度策略或者提高进程调度的优先级需要超级用户权限，否则会设置失败。  还有要注意的一点是在Docker中需要—privileged命令来获取超级用户权限后修改调度优先级，仅仅在命令行中使用sudo无效。但是还有一点无法解决——不能修改调度策略。目前还不知道怎么解决，只能回到物理机中使用Ubuntu 完成实验。 | | | |
| 结论分析与体会：  对于独立实验，完成了要求的功能，接收到SIGINT信号时优先级+1，接收到SIGTSTP信号时优先级-1（对应图中的Ctrl-C与Ctrl-Z）    结合示例程序，发现进程在调度执行的时候并不是绝对按照优先级的顺序调度。原因在于，Linux默认的调度策略是分时调度，即每个进程分配一个时间片，进程的调度是由其优先级和剩余的时间片共同决定的。优先级越小，剩余时间片越大则进程会被优先调度。较多的时间片意味着被调度的时间较少。  对于FIFO策略与RR策略来说，它们都是实时的、抢占的。当一个采用实时优先级策略的进程进入就绪态后会立即抢占非实时优先级策略的进程。当系统采用FIFO策略时，如果两个进程的优先级相同，则系统会等待先入队的进程主动放弃后才会调度下一个优先级相同的进程，而对于RR策略来说，每个任务都会执行一段时间，这样比FIFO策略要公平一些。 | | | |